

then s must be true, and he would have answered “Yes” to the first question. The son can reason in a similar way to determine that s must be true. It follows that both children answer “Yes” the second time the question is asked. ◀

1.2.6 Logic Circuits

Propositional logic can be applied to the design of computer hardware. This was first observed in 1938 by Claude Shannon in his MIT master’s thesis. In Chapter 12 we will study this topic in depth. (See that chapter for a biography of Shannon.) We give a brief introduction to this application here.

A **logic circuit** (or **digital circuit**) receives input signals $p_1, p_2, \dots, p_n$, each a bit [either 0 (off) or 1 (on)], and produces output signals $s_1, s_2, \dots, s_n$, each a bit. In this section we will restrict our attention to logic circuits with a single output signal; in general, digital circuits may have multiple outputs.

Complicated digital circuits can be constructed from three basic circuits, called **gates**, shown in Figure 1. The **inverter**, or **NOT gate**, takes an input bit p , and produces as output $\neg p$. The **OR gate** takes two input signals p and q , each a bit, and produces as output the signal $p \vee q$. Finally, the **AND gate** takes two input signals p and q , each a bit, and produces as output the signal $p \wedge q$. We use combinations of these three basic gates to build more complicated circuits, such as that shown in Figure 2.

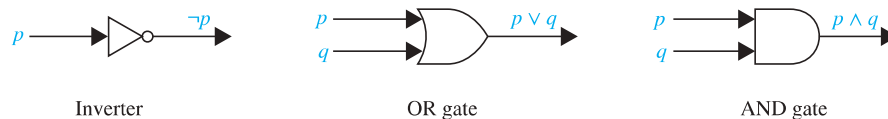


FIGURE 1 Basic logic gates.

Given a circuit built from the basic logic gates and the inputs to the circuit, we determine the output by tracing through the circuit, as Example 10 shows.

EXAMPLE 10 Determine the output for the combinatorial circuit in Figure 2.

Solution: In Figure 2 we display the output of each logic gate in the circuit. We see that the AND gate takes input of p and $\neg q$, the output of the inverter with input q , and produces $p \wedge \neg q$. Next, we note that the OR gate takes input $p \wedge \neg q$ and $\neg r$, the output of the inverter with input r , and produces the final output $(p \wedge \neg q) \vee \neg r$. ◀

Suppose that we have a formula for the output of a digital circuit in terms of negations, disjunctions, and conjunctions. Then, we can systematically build a digital circuit with the desired output, as illustrated in Example 11.

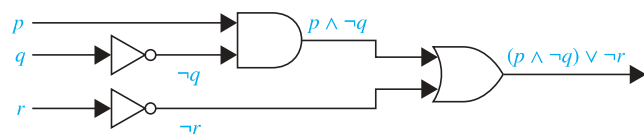


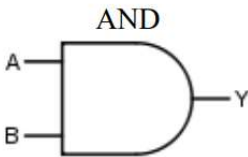
FIGURE 2 A combinatorial circuit.

In Chapter 12 we will design some useful circuits.

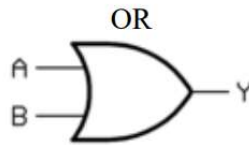
CSE1101 – Discrete Mathematics for Computer Science

Lab Sheet #1

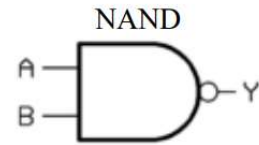
1. Fill in the truth tables of the following logic gates



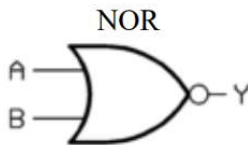
A	B	Y
0	0	
0	1	
1	0	
1	1	



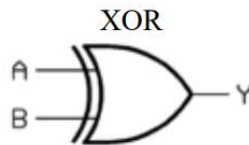
A	B	Y
0	0	
0	1	
1	0	
1	1	



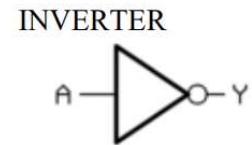
A	B	Y
0	0	
0	1	
1	0	
1	1	



A	B	Y
0	0	
0	1	
1	0	
1	1	



A	B	Y
0	0	
0	1	
1	0	
1	1	



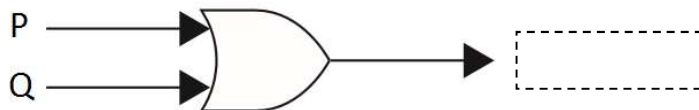
A	Y
0	
1	

2. Indicate the output of the logic gates and circuits below by filling in the blank boxes

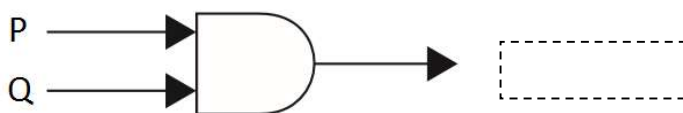
a.



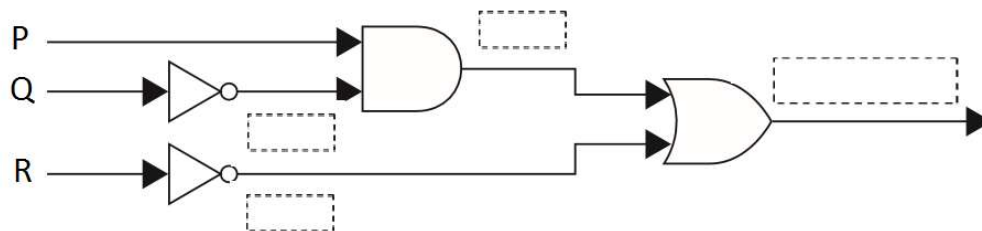
b.



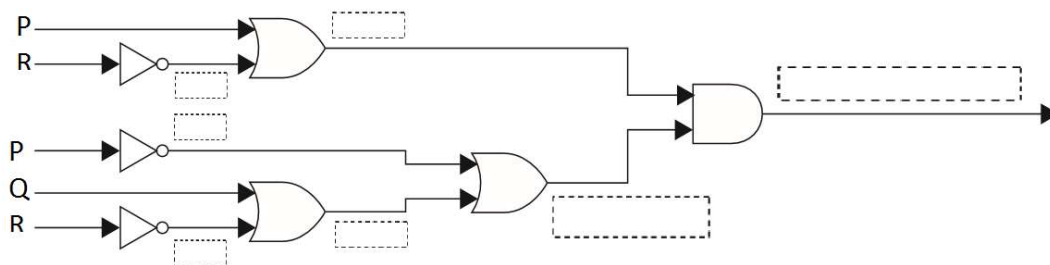
c.



d.



e.



3. Draw a circuit diagram corresponding to the following expressions:

- $(A \vee B) \wedge (B \vee C)$
- $((A \wedge B) \vee C) \wedge D$
- $(A \wedge \neg B) \vee \neg(B \vee C)$
- $\neg(A \wedge B) \vee \neg(C \wedge D)$
- $\neg(B \wedge C) \vee \neg((A \wedge B) \vee \neg C)$

End